<論文・報告>感情分析における機械学習手法の比較検討

AUTHOR(S):

中井, 諒馬

CITATION:

中井, 諒馬. <論文・報告>感情分析における機械学習手法の比較検討. ELCAS Journal 2020, 5: 23-25

ISSUE DATE:

2020-04

URL:

http://hdl.handle.net/2433/251397

感情分析における機械学習手法の比較検討

中井 諒馬

立命館守山高等学校

1 要旨

本研究ではKaggleの自然言語処理に関わるコンペティションの一つである『Toxic Comment Classification Challenge』において種々の機械学習アルゴリズムを比較することを目的とする．本データセットは感情分析に関連するものであり，感情分析とはテキストデータからテキストに含まれる感情を分析するタスクである．

本稿では機械学習を用いた感情分析の検討を行う．具体的には，感情分析をする際に1.文章をベクトル化する，2.ベクトル化したものを用いて分析を行うという二段階の方法を検討する．1ではterm frequency-inverse document frequency（tf-idf）と呼ばれるBag of Words（BoW）に基づいた方法を用い，2ではNaive Bayes Support Vector Machine（NBSVM）およびロジスティック回帰を用いた［1］．そして，これらの方法の組み合わせを比較検討した．またデータを別の言語に翻訳し元に戻すことで作られたデータの使用の有無でも比較した［2］．

2 序論

インターネットの発展とともに社会がその発展に生かすことができる情報が増えてきた反面，どのようにしてその莫大な情報を扱うのかということが重要視されている．近年のめまぐるしい計算技術の向上により機械学習が情報を処理する手段として注目を集めている．例えば，インターネット上に存在するデータには気象庁の公開する降水量や気温などの数値データ等があるが，公開データの中でも多くの情報があると考えられるものにテキストデータがある．これは，インターネットは人々がテキストによって交流をする場としても機能するからである．

本研究では『Toxic Comment Classification Challenge』において種々の機械学習アルゴリズムを比較することを目的とする．本データセットは感情分析に関連するものであり，感情分析とはテキストデータからテキストに含まれる感情を分析するタスクである．このコンペティションはWikipediaのノートページのそれぞれのコメントをtoxic, severe toxic, obscene, threat, insult, identity hateのそれぞれに該当するか否かを0～1（1に近いほど該当しているということを示す）の値を返すことで判定するモデルを構築することを目標とする．

3 研究方法

3.1 問題の定式化

今回扱う問題の入力は人があらかじめ中傷的であるかをtoxic, severe toxic, obscene, threat, insult, identity hateという観点に基づいて0または1でそれぞれ数値化してラベリングしたWikipedia上のコメントである．求められる出力はそれぞれの観点にコメントがあてはまるであろう確率である．ここでは3.2節および3.3節で述べるような手法を用いることによって与えられたコメントをd次元のベクトルxに変換し特徴量を表現しそのベクトルを分類器にかけることによってコメントがそれぞれの属性に当てはまる確率を求めることにする．

3.2 特徴量の設計

本項ではWikipediaのコメントをベクトル化して特徴量をつくるBag of Words（BoW）の一種であるterm frequency-inverse document frequency（tf-idf）について説明する．

term frequency (tf): BoWの一種であるtfでは各々のコメントに対しその中に含まれる単語の数をそのままベクトルの成分としこれを特徴量とする．

tfはterm frequencyの略で単語がある特定の文章内でどのぐらいの頻度で出現するかを表したものである．これを式で表せば文書dでの単語tのtfの値は

$$tf(t,d) = n_{t,d}$$

となり，$n_{t,d}$は単語tが文書dに現れる回数である．

しかしどのような文章においても多く出てくると思われるような単語をtfではうまく表現することができないという問題がある．

term-frequency inverse document frequency (tf-idf): tfでは，文中に多く登場する単語（例，a, the, is）をそのまま

連絡先：
mr151101@mrc.ritsumei.ac.jp（中井 諒馬，立命館守山高等学校）
myamada@i.kyoto-u.ac.jp（山田　誠，京都大学）

数えてしまうためコメントをベクトル化すると頻出単語を示す成分がどの文章においても極端に大きくなってしまうということが起こり得る．頻出単語が文章をうまく特徴付けているとは考えにくい．本問題を解決するために，頻出単語のベクトルに与える影響を弱めつつ，ある文章に固有で珍しい単語の情報を大きく特徴量ベクトルに反映することをおこなうことのできるtf-idfを利用する．tf-idfはそれぞれの文章, 単語ごとにtfとidfという値を計算し掛け合わせることで得ることができる．

idfはinverse document frequencyの略でありこれは与えられた全ての文章のうちどれだけの文章がその単語を含んでいるかを示す量である．idfが大きいほどその単語が様々な文章で現れることがないということを示し，もしその単語が文章中に現れた際には大きな情報を持つことになる．この値はある特定の文章を優遇するということはないため個々の文章に依存せず単語ごとに定まる．厳密には単語tに対し以下で定義される．

$$idf(t) = \log \frac{1+n}{1+df(t)} + 1$$

ここでnは与えられた文書の総数を示し$df(t)$はn個の文書のうちの何個が単語tを含むのかを示す．

そしてtf-idfは次のように定義される

$$tf - idf(t,d) = tf(t,d)idf(t)$$

このようにして求められるtf-idfを成分として各文章をベクトル化する．

3.3 分類器

今回は比較対象としてロジスティック回帰とNBSVMを採用した．ここではこれらについて簡潔に説明を加えることにする．

ここでは$\boldsymbol{x}_i \in \mathbb{R}^d$をtf-idfで変換された文書のベクトルとする．

ロジスティック回帰ではロジスティック関数を用いて以下のように予測値y_iの確率を定義する．

$$P(y_i \mid \boldsymbol{x}_i) = \frac{1}{1+\exp\left(-(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\right)}$$

ここで$\boldsymbol{w} \in \mathbb{R}^d$および$b \in \mathbb{R}$は学習により求まるパラメータである．ロジスティック回帰では損失関数を定義し，パラメータを最適化する．[3]Cは正則化の強さを指定するハイパーパラメータである．

$$L(\boldsymbol{w},b) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_{i=1}^{n}(\ln(\exp - y_i(\boldsymbol{x}_i{}^\top \boldsymbol{w} + b)) + 1)$$

NBSVMの分類器ではJeremy Howard氏が書いたソースコードを使わさせて頂いた［4］.

3.4 実験設定

今回の実験ではKaggleのToxic Comment Classification Challengeで提供されている英語のデータ（データAと呼ぶことにする）とコメントを一度他言語（ドイツ語, フランス語, スペイン語）に翻訳し英語に翻訳し直したデータ（データBと呼ぶことにする）を用いた．

データAは65535個のコメントとそれに対して人がつけたラベルからなっている．データBはデータAの３倍である196605個のコメントを含んでいる．

A, Bそれぞれにトークン処理を施した後にコメントをベクトル化する際にはtf-idfを用いた．これにはscikit-learnで提供されている関数を使用した．それぞれの手法においてngram_rangeのパラメーターを(1,1),(1,2),(1,5)として比較した．

その他のパラメーターは同一で以下の通りである．

```
TfidfVectorizer
min_df=3, max_df=0.9, strip_accents='unicode'
use_idf=1, smooth_idf=1, sublinear_tf=1

CountVectorizer
min_df=3, max_df=0.9, strip_accents='unicode'
```

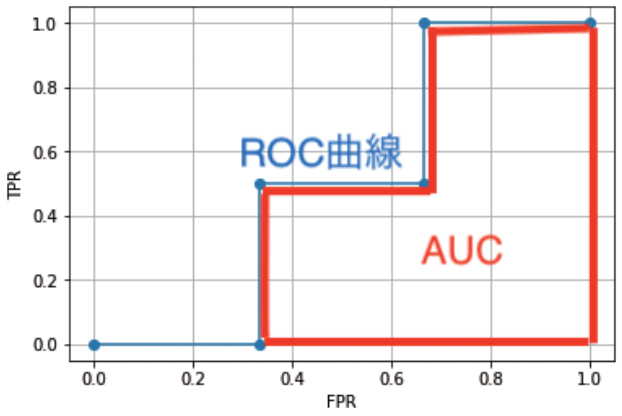
LogisticRegression（C = 4, dual = True）としてある

3.5 評価の方法

Kaggleでの提出結果の評価はtoxic, severe toxic, obscene, threat, insult, identity hateの項目のそれぞれについて求めたAUCの平均値で与えられる．

今ある項目Fについてのi番目のコメントの確率の出力の結果がp_iだったとする．この確率が与えられた際にそのコメントが実際にFに該当するか否かを判断するときに基準となる確率を閾値（qとする）と呼ぶ．つまり$p_i >= q$のときi番目のコメントがFに該当すると考えるということである．あるコメントがFに当てはまっているということをそのコメントは陽性であるということにしその逆を陰性ということにすると閾値を定めることにより実際に陽性で予測も陽性となるコメントと実際には陰性であるにも関わらず陽性と予測されてしまうコメントが現れる．

実際は陰性であるコメントを誤って陽性であると判断してしまった割合をFPR（偽陽性率）といい実際に陽性であるコメントを正しく陽性と推定できたものの割合をTPR（真陽性率）という．FPR，TPRは閾値qにより決まり前者は小さく後者は大きいと性能の良い予測ができているということになる．閾値qを動かした時に（FPR, TPR）が動く曲線をROC曲線といいROC曲線の下の部分の面積のことをAUCという．AUCが大きいほど予測モデルがうまくできていることになる．



4 結果

以下の表のようなスコアが得られた.

score table	NBSVM	logistic regression
tfIDF(n_gram = (1,1))	0.97468	0.97549
tfIDF(n_gram = (1,2))	0.97722	0.97376
tfIDF(n_gram = (1,5))	0.97663	0.97074
BoW(n_gram = (1,1))	0.93941	0.93994
BoW(n_gram = (1,2))	0.93980	0.94194
BoW(n_gram = (1,5))	0.93633	0.93978

以下の表のようなスコアが得られた.

score table	NBSVM	logistic regression
tfIDF（A）	0.97722	0.97376
tfIDF（A+B）	0.97766	0.97468
BoW（A）	0.93980	0.94194
BoW（A+B）	0.92819	0.94601

5 考察

上記結果にあげた表から他の値よりもn-gram = (1,2)とした方が, BoWよりtf-idfを用いた方が, ロジスティック回帰よりNBSVMを用いた方が, データAだけよりもデータA+Bを用いた方が良いパフォーマンスを発揮するのではないかと考えられる.

tf-idfでは各単語の文章におけるユニークさを評価に入れているのでBoWよりも文章を適切にベクトル化できているのではないかと考えられる. データを水増しすることで精度があがったと考えられる.

6 謝辞

この研究を行うにあたり京都大学大学院情報学研究科山田研究室の皆さんにお世話になりました. ありがとうございました.

References

［1］Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

［2］https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/48038.

［3］https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.

［4］https://www.kaggle.com/jhoward/nb-svm-strong-linear-baseline.